Putting telemetry and events on the web

5 October 2006

Putting telemetry and events on the web

This gives some information on making ITOS events and telemetry pages available on the web.

On the web server

1. Some publicly accessible directory, typically /home/*mission*/public_html, should contain three symbolic links:

   classes points to $ITOS_DIR/classes.

   pages points to the root of a page directory tree. Both text-based pages, those with .page suffixes, and graphics pages, those with .disp suffixes, can be displayed.

   tcvol2 points to the directory containing the html files created by ⟨undefined⟩ [dbxodb], page ⟨undefined⟩

2. An html page in that directory, typically Welcome.shtml, should contain links to:

   classes/evtdsp.cgi
   to display events

   classes/select_page.cgi?$ITOS_DIR+mission
   to display telemetry pages in the tree pointed to by the *pages* link.

The group directory gives more information about how mission home directories are normally set up, and contains an example Welcome.shtml file.

If ITOS is running on the web server, that's all there is to it.

When ITOS is not running on the web server

If ITOS is not running on the web server, application evtforward must be started on the web server to get events from the main ITOS machine and make them available to web applets. That can usually be done with the following command at a shell prompt on the web server machine:

```
$ $ITOS_DIR/bin/itos_getevt itos_host
```

where *itos_host* is the name of the main ITOS machine.

For this to work:

1. Environment variable $ITOS_DIR must be set.
2. Directories $ITOS_DIR/bin and $ITOS_DIR/classes on the web server must contain the usual ITOS files.
3. Evtforward must be able to make a socket connection from the web server to the main ITOS machine.

Evtforward continues to run even when ITOS is shut down. It reconnects when ITOS is restarted. Shutting evtforward down requires a kill command at a shell prompt.

When ITOS is inside a firewall

1. A hole is made in the firewall for events. The hole is for a TCP socket to the web server, port number *event_port*.

2. On the web server, outside the firewall, $ITOS_DIR/bin/itos_getevt is modified so it starts evtforward with this command:

   `java evtforward -protectedsource $1` *event_port* `&`

   That tells evtforward that it cannot connect to the ITOS machine as usual. Instead it will open a server socket on port *event_port* and wait for the event source to connect to it.

   After that modification, any running evtforward should be killed and itos_getevt run to start a new version:

   `$` *$ITOS_DIR*`/bin/itos_getevt` *itos_host*

3. On the ITOS machine, script $ITOS_DIR/bin/itos is edited to modify the way it starts its evtforward. The usual command is:

   `java evtforward -source $ITOS_TMCTRL_HOST -wait -checkServer &`

   That is modified to:

   `java evtforward -source $ITOS_TMCTRL_HOST -wait -checkServer -getclient` *web_host ev*

   where *web_host* is the name of the web server machine. That tells evtforward that a client is not able to connect in the usual manner. Instead evtforward on the ITOS machine will connect to the server socket opened by evtforward on the web server.

   After that modification, any running evtforward in the ITOS machine should be killed. A new instance will be started when ITOS is started.

Evtforward command line arguments describes all evtforward arguments. Evtforward gives general information on evtforward.